

Capítulo 7: Document Object Model (DOM)

Trindade da Web

HTML



CONTENT

NOUNS

```
<p></p>
```

means "paragraph"

CSS



PRESENTATION

ADJECTIVES

```
p {color: red;}
```

means "the paragraph
text is red"

JS

DYNAMIC EFFECTS/
PROGRAMMING

VERBS

```
p.hide();
```

means "hide the
paragraph"

World Wide Web (Web)

- A World Wide Web (ou Web) é um espaço de informações construído sobre a Internet. Os recursos da Web são acessíveis por meio de sua *URL* e podem conter *hiperlinks* para outros recursos.
 - Por isso recursos da Web são também chamados de hipertexto
- Um documento da web é chamado *página web*. A criação de páginas web geralmente envolve três tecnologias:
 - HTML, para estruturar o conteúdo,
 - CSS, para definir sua apresentação
 - JavaScript, para adicionar interatividade

HTML

Um documento HTML é feito de texto e elementos estruturais, chamados *tags*, que descrevem o conteúdo da página, como:

- Parágrafos,
- cabeçalhos,
- hiperlinks,
- imagens, etc.

ps1: um bom recurso para se estudar HTML é o site [w3schools.com](https://www.w3schools.com)

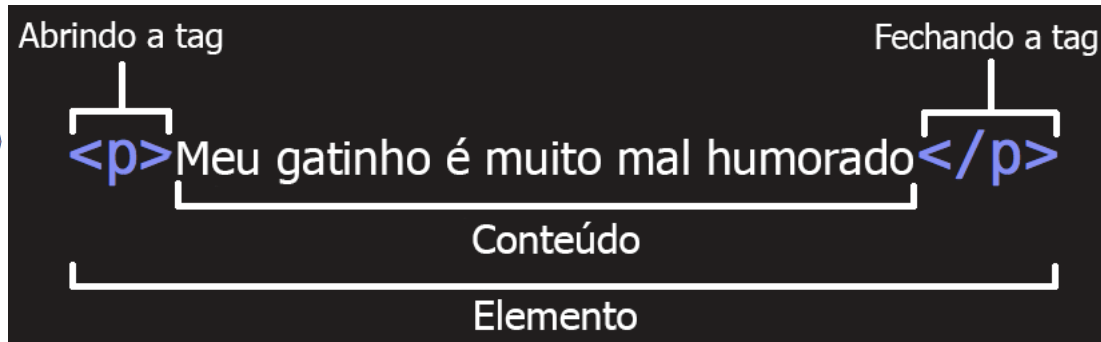
```
<html>
  <head>
    <meta charset="utf-8">
    <title>Minha página Web</title>
  </head>
  <body>
    <h1 class="centralizado">Minha página Web</h1>
    <p id="negrito" class="centralizado">Olá! Meu Nome é César</p>
    <p>Moro em <a href="https://en.wikipedia.org/wiki/Fortaleza">Fortaleza</a>. </p>
  </body>
</html>
```

ps2: essa indentação normalmente se vê em documentos HTML é apenas uma ajuda visual para humanos

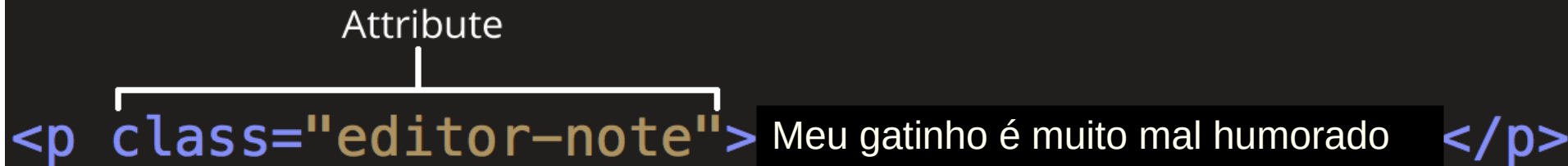
Anatomia de um elemento HTML

Um elemento é composto pela tag de abertura, tag de fechamento e o conteúdo*:

Elementos também podem conter *atributos*, que contém informação extra sobre o elemento:



*Alguns elementos não possuem nem conteúdo, nem tag de fechamento (ex. ``, `
`, `<input>`, etc)



Algumas tags comuns

Cabeçalhos:

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

Links:

`texto do link`

Imagens:

``

Formulários:

`<form>`

elementos do formulário

`</form>`

Exemplo:

`<h1>Bem vindo ao IFCE!</h1>`

`IFCE`

``

`<form action="/action_page.do">`

`<label for="nome">Nome:</label> <input type="text" id="nome" name="nome">
`

`<label for="email">email:</label> <input type="text" id="email" name="email">

`

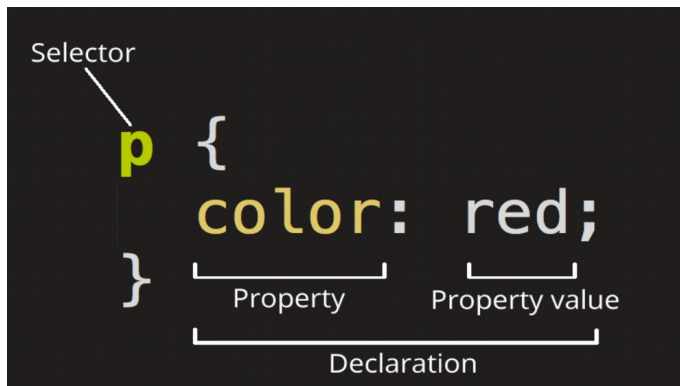
`<input type="submit" value="Submit">`

`</form>`

CSS

CSS, ou Cascading Style Sheets, é uma linguagem usada para alterar a apresentação de páginas da web.

O CSS usa *seletores* para declarar a quais elementos HTML um estilo se aplica. Os elementos podem ser selecionados por nome de tag (h1), por classe (.laranja) ou por identificador (#negrito).



```
/* Todos elementos h1 são fundo preto e letra branca */
```

```
h1{
```

```
  color: white;
```

```
  background-color: black;
```

```
}
```

```
/* Todos elementos com a classe "laranja" são da cor laranja */
```

```
.laranja{
```

```
  background-color: orange;
```

```
}
```

```
#negrito{
```

```
  font-weight: bold;
```

```
}
```

Seletores CSS

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Minha página Web</title>
```

```
</head>
```

```
<body>
```

```
<h1 class="centralizado">Minha página Web</h1>
```

```
<p id="negrito" class="centralizado italico">Olá! Meu Nome é César</p>
```

```
<p>MorO em <a href="https://en.wikipedia.org/wiki/Fortaleza">Fortaleza</a>. </p>
```

```
</body>
```

```
</html>
```

```
p {  
  color: red;  
}  
#negrito {  
  font-weight: bold;  
}  
.centralizado {  
  text-align: center;  
}  
.italico {  
  font-style: italic;  
}
```


Exercício

1) Crie sua página de apresentação pessoal com a seguinte estrutura:

Título:

Seu nome

Parágrafo:

pequena apresentação sobre você

Foto:

Insira uma foto sua

Informações adicionais

Link para página externa (de sua cidade, p.ex.)

2) Após pesquisar sobre os seguintes elementos, incrementar sua página de apresentação pessoal com cada uma deles:

Listas (ordenadas e não ordenadas)

Tabelas

Mídia (vídeos e áudios)

3) Estilizar sua página a gosto

Inserindo o CSS em um documento HTML

Existem três maneiras de inserir uma folha de estilo CSS:

CSS Externo:

arquivo externo referenciado no elemento `<link>` que deve ficar na seção `<head>`

CSS Interno:

definido dentro do elemento `<style>`, dentro da seção `<head>`.

usada se uma única página HTML tiver um estilo exclusivo

CSS Inline:

adicione o atributo *style* aos elementos desejados.

O atributo *style* pode conter qualquer propriedade CSS.

Exemplos:

```
<link rel="stylesheet" href="meuEstilo.css">
```

```
<style>
```

```
h1 { color: maroon; margin-left: 40px; }
```

```
</style>
```

```
<p style="color: red;">Meu parágrafo! </p>
```

E onde entra o Javascript? - DOM!

A API DOM permite-nos fazer inúmeras operações programaticamente:

- identificar e seleccionar elementos
- manipular elementos (CRUD da DOM)
- aplicar estilo
- adicionar eventos
- etc.

DOM

- Javascript, como sabemos, é uma linguagem orientada a objetos
- Assim, a maneira de interagir com elementos de uma página HTML é criando objetos que emulem tais elementos e métodos que permitam manipulá-los:

HTML

```
<p>  
  Look at this octopus:  
    
  Cute, huh?  
</p>
```

DOM Element Object

Property	Value
tagName	"IMG"
<u>src</u>	"octopus.jpg"
alt	"an octopus"
id	"icon01"

JavaScript

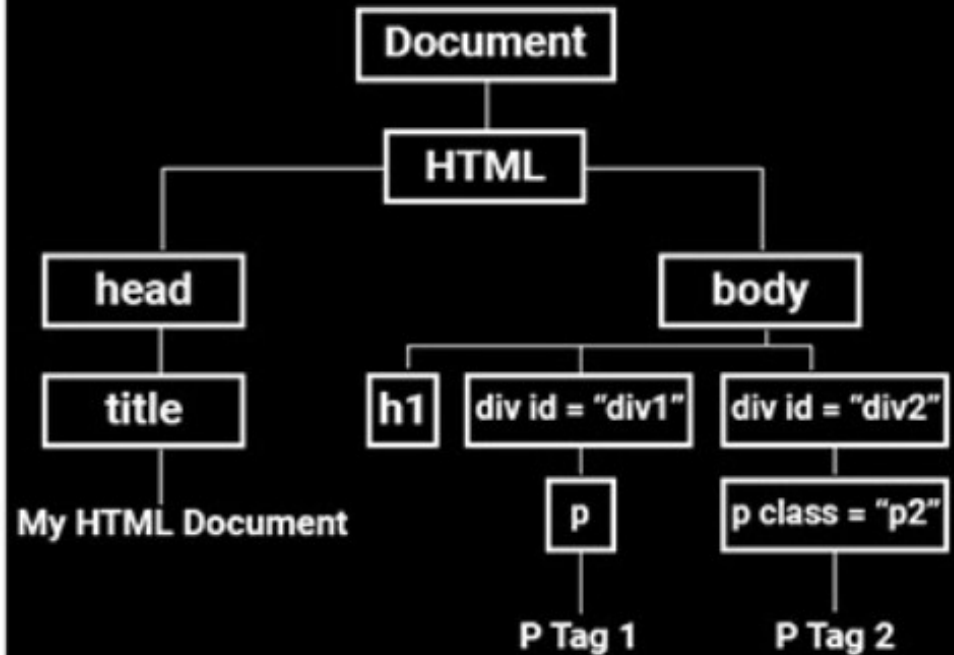
```
var icon = document.getElementById("icon01");  
icon.src = "kitty.gif";
```

HTML vs. DOM

HTML Document

```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```

Document Object Model (DOM)



DOM: Manipulando os elementos

Neste capítulo veremos métodos básicos da DOM, mas que nos permitirão executar tarefas importantes, como:

- 1) Encontrar os elementos a serem manipulados. Podemos buscar elementos por id, nome do tag, nome da classe, seletores CSS, entre outros.
- 2) Adicionar novos elementos
- 3) Modificar elementos existentes
- 4) Remover elementos
- 5) Modificar estilo de um elemento
- 6) Trabalhar com eventos

DOM: Selecionando os elementos

Podemos encontrar elementos por meio de diversos métodos. Todos esses métodos pertencem ao objeto *document*, que representa o elemento `<html>` :

```
const pElem = document.getElementsByTagName('h1')    //seleciona pelo nome do tag
```

```
const pElem = document.getElementById("intro")      //seleciona pelo id (único elemento!)
```

```
const pElem = document.getElementsByClassName('centralizado') //seleciona pelo nome da classe
```

```
const pElem = document.querySelector("#negrito")    //seleciona único elemento via seletor CSS
```

```
const pElem = document.querySelectorAll(".centralizado") //seleciona vários elementos via seletor CSS
```

DOM: Adicionando novos elementos

Para criar elementos, usamos `createElement()`, também de *document*

Uma vez criado o elemento, devemos inseri-lo no elemento “pai”

```
let pElem = document.createElement('p'); //cria elemento “vazio”
```

```
pElem.textContent = 'Este é um novo parágrafo adicionado'; //insere conteúdo
```

```
document.body.appendChild(pElem); //insere no pai (nesse caso, <body>)
```

```
document.body.insertBefore(pElem, elemExistente)//se não quiser inserir no fim
```


DOM: Modificando elementos

Para modificar o conteúdo de um elemento, pode-se usar os métodos:

- `innerHTML` (interpreta o HTML)
- `innerText` (trata HTML como uma string qualquer)

```
<p id="p1">Alô, mundo!</p>
```

```
<script>
```

```
    document.getElementById("p1").innerHTML = "<strong>Olá, Universo</strong>";
```

```
</script>
```

DOM: Removendo elementos

Também podemos remover elementos da DOM. Para tal, basta chamar o método `remove()` diretamente no elemento a ser removido*:

*O método `remove()` não funciona em navegadores mais antigos. Nesse caso, usa-se `removeChild()`, que requer o elemento pai:

```
pai.removeChild(filho)
```

```
<div>  
  <p id="p1">Este é um parágrafo.</p>  
  <p id="p2">Este é outro parágrafo.</p>  
</div>
```

```
<script>  
  const e = document.getElementById("p1");  
  e.remove();  
</script>
```

DOM: Modificando o estilo de um elemento

O HTML DOM permite que o JavaScript altere o estilo dos elementos HTML.

Para alterar o estilo de um elemento HTML, use a seguinte sintaxe:

```
document.getElementById (id).style.property = "valor"
```


```
<p id="p2">Alô, mundo!</p>
```

```
<script>
```

```
document.getElementById("p2").style.color = "blue";
```

```
</script>
```

*neste exemplo, a
propriedade que se
deseja modificar é a cor*

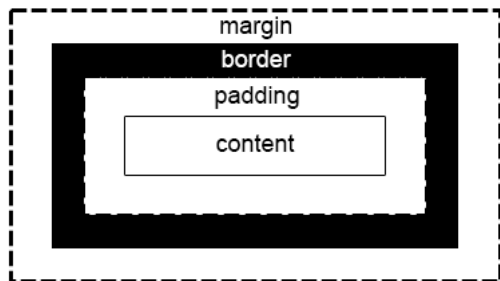


DOM: Aplicando estilo

Aplica-se estilo a elementos da DOM da seguinte maneira:

```
document.getElementById(id).style.propriedade =  
"novo estilo"
```

A tabela ao lado exemplifica algumas propriedade comuns e a figura abaixo mostra o modelo de caixa (box model) do CSS:



Propriedade	DOM	CSS
Cor	color	color
Preenchimento	padding	padding
Cor de fundo	backgroundColor	background-color
Largura da borda	borderWidth	border-width
Tamanho da fonte	fontSize	font-size
Família da fonte	fontFamily	font-family
Margem	margin	margin

Juntando HTML, CSS e Javascript

Um documento HTML pode incluir uma folha de estilo CSS por meio da tag <link> e um arquivo JavaScript com uma tag <script>.

```
<html>
  <head>
    <!-- Info sobre a página: título, charset, etc -->
    <!-- Link para o CSS -->
    <link href="estilos/meuEstilo.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    <!-- Conteúdo da página -->
    <!-- Link para script JS -->
    <script src="meuPrograma.js"></script>
  </body>
</html>
```

Exercícios DOM

```
<html>
<head>
  <meta charset="utf-8">
  <title>Minha página Web</title>
</head>
<body>
  <h1 class="centralizado">Minha página Web</h1>
  <p id="negrito" class="centralizado italico">Olá! Meu Nome é
César</p>
  <p>MorO em <a
href="https://en.wikipedia.org/wiki/Fortaleza">Fortaleza</a>. </p>
</body>
</html>
```

- 1) Personalize o template ao lado com seus dados
- 2) Liste 3 maneiras de mudar a cor da frase com seu nome para amarelo: usando o nome da tag, o id e a classe do elemento
- 3) Acrescente uma nova div e dentro dela um parágrafo com um texto sobre você. Estilize a seu gosto.
- 4) Remova essa nova div.